

Local Redesigning of Airspace Sectors

Irina Kostitsyna* and Joseph Mitchell†

Abstract. In this paper we study the *Airspace Sectorization Problem* (ASP) where the goal is to find an optimal partition (sectorization) of the airspace into a certain number of sectors, each managed by an air traffic controller. The objective of the ASP is to find a “well-balanced” sectorization that distributes the workload evenly among the controllers. We formulate the ASP as a partitioning problem of a set of moving points in a polygonal domain. In addition to the requirement of balancing the workload, we introduce restrictions on the geometry of the sectorization which come from the Air Traffic Management aspects.

We investigate several versions of the problem that arise from different definitions of the notion of the workload and various choices of geometric restrictions on the sectorization. We conclude that most of the formulations of the problem, except maybe in some trivial cases, are NP-hard.

Finally, we propose a *Local Redesigning Method* (LRM), a heuristic algorithm that rebalances a given sectorization by adjusting the boundaries of the sectors. We evaluate LRM experimentally on synthetically generated scenarios as well as on the real historical traffic data. We demonstrate that the sectorizations produced by our method are superior in comparison with the current sectorizations of the US airspace.

1 Introduction

The current design of the *National Airspace System* (NAS) was developed based on flight routes that were formed historically. Over the years, the demand and the geometry of the routes have changed dramatically, yet the NAS has undergone little change. As a consequence, the current sectorization is no longer able to accommodate the rapidly increasing demand.

The problem of designing a flexible and dynamic airspace architecture that is able to adapt to changing traffic flows is addressed by the *Dynamic Airspace Configuration* (DAC) project as a part of the *Next Generation Air Transportation System* (NextGen) [1]. The NAS is partitioned into 22 Air Route Traffic Control Centers, each subdivided into sectors, which are overseen by air traffic controllers. The maximum workload that air traffic controllers can safely handle results in a limitation on the capacities of the sectors. If the changing traffic causes the demand on a sector to exceed its capacity, the sector will not be able to accommodate all the incoming flights. This will lead to some flights being rerouted around the congested area, and others to be delayed.

There are two basic approaches to handling changes in traffic. The first one is to design a new sectorization from scratch. Such methods concentrate on new traffic patterns, while discarding the old sectorization. Examples of this approach include: a cell-based Mixed Integer Programming (MIP) model [2,3]; a sectorization method using binary space partitions [4,5]; a Voronoi diagram method [6]; a graph partitioning method [7], etc. While a clean-sheet sectorization design provides a wide range for finding an optimal solution, it is undesirable due to a human factor; it is important for controllers to be familiar with the geometry of sectors and traffic patterns.

The second approach is to perform a local rebalancing of the existing sectorization without introducing dramatic changes to the sectors. Klein *et al.* [8] present an algorithm that shifts pre-specified thin subsectors between adjacent sectors to rebalance them. A local method for adjusting the boundaries of sectors that provides “outs” (or extra space) around weather constraints is proposed by Drew [9]. This method uses a force-based approach to adjust the boundaries in order to improve the capacities of the sectors that are most impacted by weather. The Voronoi method presented by Xue [6] includes a local rebalancing option as well

* Computer Science Department, Stony Brook University, ikost@cs.stonybrook.edu

† Department of Applied Mathematics and Statistics, Stony Brook University, jsbm@ams.sunysb.edu

as a clean-sheet design option; it adjusts the design of sectors by iteratively moving the Voronoi centers. The existing sectorization (the one created by applying the Voronoi method in clean-sheet mode) is used as the seed for a genetic algorithm that adapts the sectorization to the new demand. Local methods may change the topology of the design, including changes in the number of sectors. For instance, a pair of adjacent sectors may be combined into one, or a single sector may be split into two. Sector combination methods, based on computing predicted capacity gaps and then greedily combining pairs of sectors having the largest such gaps, have been proposed and examined in experiments of [10].

In this paper we discuss theoretical aspects of the ASP and present a new approach to the problem of re-designing sectorizations by local adjustments of the sector boundaries. We present a *Local Redesigning Method* (LRM), a highly customizable multi-criteria optimization heuristic. We have developed GEOSECT-LOCAL (a complement to GEOSECT sectorization tool introduced in [4, 5]) that performs LRM adjustments to an input sectorization. The input sectorization can be any partition of the airspace region of interest, including current NAS sectorization, manually entered sectors, or the output of any other sectorization method, such as the top-down GEOSECT clean-sheet method.

2 Problem Statement

The main objective of the ASP is to find a sectorization that distributes the workload evenly among the sectors. In this study we use three different metrics for the workload: $AC_{max}(\sigma)$ —the maximum aircraft count in sector σ , $AC_{avg}(\sigma)$ —the time-average aircraft count in σ , and $\delta(\sigma)$ —the delay introduced by the overload of σ . We give precise definitions of AC_{max} , AC_{avg} and δ in Section 3.2.

As discussed in [5], human factor should also be taken into account when designing a sectorization. This leads to geometric restrictions that can be roughly divided into two groups: flow conformance requirements and sector geometry requirements. Flow conformance requirements reflect the compatibility of the sector boundaries with the traffic flows, weather obstacles, and locations of the airports or other singular points (refer to Figure 1). Airplanes should pass far enough from a sector boundary, and when they cross the boundaries between sectors the intersection angle should be nearly orthogonal. Furthermore, there is a restriction on the minimum dwell time for an airplane: after entering a sector the airplane should spend some time within it before exiting. Any *critical point*, such as an airport or a conjunction/intersection of major flows, should also be well inside the sector to give air traffic controllers time to safely manage possible conflicts. If there is a weather obstruction in a sector, there should be enough of the throughput capacity in the directions of traffic flows to allow the sector to accommodate them. The second group of requirements regulates the geometry of sectors: we require the sectors to be convex or nearly convex and bound minimum and maximum angles.

Now, to formally state the ASP we can choose any of the discussed requirements to define the objective function $f(\cdot)$, and others to construct a set of constraints \mathbb{C} . For example, we can choose to optimize the average aircraft count subject to constraining sectors convexity and maximum aircraft count:

$$\begin{aligned} &\text{optimize } f = \max_{\sigma} AC_{avg}(\sigma) \\ &\text{subject to constraints } \mathbb{C} = \{\text{convex sectors}, AC_{max}(\sigma) \leq AC_{max}^*\}. \end{aligned}$$

Problem 1 (ASP). Given a polygonal domain \mathcal{D} , a set of aircraft trajectories \mathcal{T} in a time interval $[0, T]$, and, possibly, a set of dominant flows \mathcal{DF} , a set of critical points \mathcal{CP} , and a set of weather obstacles \mathcal{W} , find a partition \mathcal{S} of \mathcal{D} into k sectors to optimize $f(\mathcal{S}, \mathcal{T}, \mathcal{CP}, \mathcal{DF}, \mathcal{W})$ subject to constraints \mathbb{C} .

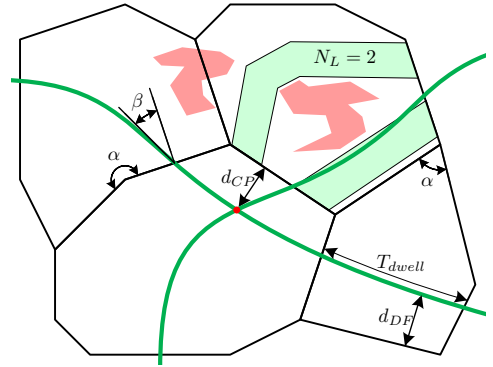


Fig. 1: Geometry of sectors, traffic flows

We also consider a dual problem whose goal is to minimize the number of sectors in a sectorization subject to a set of constraints.

Problem 2 (dual ASP). Given a polygonal domain \mathcal{D} , a set of aircraft trajectories \mathcal{T} in a time interval $[0, T]$, and, possibly, a set of dominant flows \mathcal{DF} , set of critical points \mathcal{CP} , and a set of weather obstacles \mathcal{W} , find a partition \mathcal{S} of \mathcal{D} into minimum number of sectors subject to constraints \mathbb{C} .

2.1 Complexity of ASP

Few special cases of ASP can be solved polynomially. Consider ASP whose objective is to minimize the AC_{avg} . In the case when the airspace region is a convex polygon, this problem can be solved, for example, by line-sweeping technique. Balancing AC_{avg} is equivalent to balancing the total length of aircraft trajectories in each sector. Sweep the line perpendicularly, slicing the polygon each time when a sector becomes “full”. If the polygon is non-convex, this can result in sectors having multiple disconnected components, but this issue can be easily worked around by connecting the pieces with thin corridors along the polygon boundary.

Basu *et al.* [4] prove that ASP whose objective is to minimize the maximum aircraft count with axis aligned rectangular sectors is NP-hard. Farrahi *et al.* [11] strengthen this result by showing it to be NP-hard to optimize the maximum aircraft count version of the ASP with no geometric restrictions on sectors by reduction from the Planar-Partition-into-Triangles, or PLANAR-P3. They also show NP-hardness of the ASP version with the objective to minimize the total number of tracks in each sector, as well as the version with the objective to minimize the maximum number of aircrafts in a sector during any N -minute time interval (with a constant N).

In this paper we show that a number of special cases of ASP are weakly NP-hard by a simple reduction from the PARTITION problem. The construction of the proof is basically the same for all the cases with slight differences that account for the types of the constraints. Here we show the proof that the ASP with the constraint on the distance between critical points and sector boundaries with the objective to minimize AC_{avg} or AC_{max} is weakly NP-hard. The PARTITION problem asks to determine if it is possible to divide a set of positive integers a_1, a_2, \dots, a_n into two subsets such that the total sums of the numbers in each subset are equal. Given an instance of the PARTITION problem we construct an instance of the ASP with the required number of sectors $k = 2$ (refer to Figure 2). Any solution of the PARTITION problem corresponds to a subdivision of the airspace into two sectors that solves the ASP. Thus, we have the following theorem:

Theorem 1. *ASP with the constraint on the distance between critical points and sector boundaries with the objective to minimize AC_{avg} or AC_{max} is weakly NP-hard.*

Theorem 2. *ASP with the constraint on the minimum dwell time with the objective to minimize AC_{avg} or AC_{max} is weakly NP-hard.*

Theorem 3. *ASP with the constraint on the intersection angles of dominant flows and sector boundaries with the objective to minimize AC_{avg} or AC_{max} is weakly NP-hard.*

Theorem 4. *ASP with the objective to minimize AC_{max} is weakly NP-hard.*

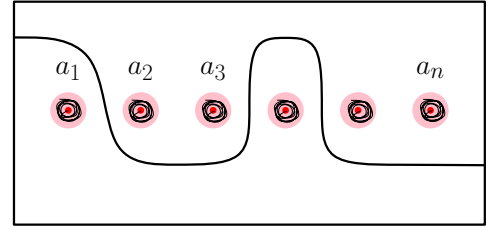


Fig. 2: An instance of the ASP with the constraint on the distance between critical points and sector boundaries constructed from an instance of the PARTITION problem. Red points represent airports with light-red disks showing the areas prohibited for sector boundaries to cross. Put a_1, a_2, \dots, a_n number of aircrafts to circle around the airports within the “no-crossing” areas.

3 Overview of the Local Redesigning Method

As we have seen, solving the ASP optimally is difficult. We have developed a heuristic LRM that improves a given sectorization by locally adjusting sector boundaries. To estimate the “quality” of sectors of a sectorization before and after an adjustment, the LRM introduces an objective function $\text{cost}(\sigma, \mathcal{T}, \mathcal{DF}, \mathcal{CP}, \mathcal{W})$ that depends on the constraints of the ASP. Here σ is a sector, \mathcal{T} is a set of aircraft trajectories, \mathcal{DF} is a set of dominant flows, \mathcal{CP} is a set of critical points, and \mathcal{W} is a set of weather obstacles. We will denote the objective function as $\text{cost}(\sigma)$ for short. The value of $\text{cost}(\sigma)$ is zero if the constraints are satisfied, and it grows if the constraints get violated. We have extracted a set of elemental parameters, each of which can numerically measure some simple property of a sector in a given sectorization. These parameters include the maximum and time-average aircraft count, the estimated delay, the angles of sectors, the distances between critical points and sector boundaries, the intersection angles of traffic flows with the boundaries, etc. For each of the ASP constraints and a corresponding parameter we define a simple cost function that determines a penalty for the violation of the given constraint, the more the violation, the higher the penalty. We take a linear combination of these cost functions to be the objective function in the LRM:

$$\text{cost}(\sigma) = \sum_{c \in \mathcal{C}} w_c \text{cost}_c(\sigma), \quad (1)$$

where c is a constraint, $\text{cost}_c(\sigma)$ is a cost function associated with c , and w_c is a user-specified constant. Now the LRM can optimize an input sectorization with respect to this objective function. We will describe the parameters and the cost functions in more details in Section 3.2.

At a high level, LRM redesigns a given sectorization with a sequence of local adjustments, each making small changes to the geometry of the sectors, *e.g.*, by moving (or inserting/deleting) vertices or edges (see Section 3.1). Notice, that our method can be applied to 3D sectorizations as well, but here we focus on sectorizations that are constant with the altitude, so the problem can be viewed as planar partitioning.

Consider a sectorization, \mathcal{S} , a planar partition of an airspace into sectors $\{\sigma_1, \sigma_2, \dots, \sigma_k\}$, where each σ_i is a simple polygon. The boundary $\delta\mathcal{S}$ of \mathcal{S} is the boundary of the region of interest. LRM adjusts the interior of \mathcal{S} leaving $\delta\mathcal{S}$ unchanged. Let $\mathcal{L}(\mathcal{S})$ be a set of feasible local adjustments to sectorization \mathcal{S} (the way these are selected is described in Section 3.1).

Input: Sectorization \mathcal{S} and traffic \mathcal{T} ; and optionally set of dominant flows \mathcal{DF} , set of critical points \mathcal{CP} , and weather obstacles \mathcal{W}

Output: Sectorization \mathcal{S}^* which is in a local minimum with respect to the objective function $\text{cost}(\sigma, \mathcal{T}, \mathcal{DF}, \mathcal{CP}, \mathcal{W})$

```

1: loop
2:   define  $\text{cost}(\sigma) = \text{cost}(\sigma, \mathcal{T}, \mathcal{DF}, \mathcal{CP}, \mathcal{W})$ 
3:    $\mathbb{S} \leftarrow \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ , sorted by  $\text{cost}(\sigma)$ 
4:   loop
5:     if  $\mathbb{S}$  is empty then
6:       return  $\mathcal{S}$ 
7:     end if
8:      $\sigma_{max} \leftarrow \mathbb{S}.\text{pop}()$ 
9:      $c_0 \leftarrow \text{cost}(\sigma_{max})$ 
10:     $c_\ell \leftarrow \min_{\ell \in \mathcal{L}(\mathcal{S})} \max_{\sigma \in \mathbb{S}} (\text{cost}(\sigma_\ell))$ 
11:    if  $c_\ell > c_0$  then
12:      apply  $\ell$  to  $\mathcal{S}$ 
13:      go to 1
14:    end if
15:  end loop
16: end loop
```

Fig. 3: Local Redesigning Method optimizes sectorization \mathcal{S} with respect to the multi-criteria objective function. Here $\mathcal{L}(\mathcal{S})$ is a set of feasible local adjustments to \mathcal{S} at every step of the algorithm, and σ_ℓ is a sector σ after applying the local adjustment ℓ .

The objective function associates a penalty, $\text{cost}(\sigma)$, to each sector σ . At each iteration of the main loop, we examine the sector, σ_{max} , with the highest cost. Consider $\mathcal{L}(\sigma_{max}) \subset \mathcal{L}$: all the local adjustments that affect sector σ_{max} . LRM selects the best local adjustment $\ell_{max} \in \mathcal{L}(\sigma_{max})$ that maximizes its “benefit”, *i.e.*, it minimizes the maximum cost among the sectors affected by $\ell(\sigma_{max})$, including σ_{max} itself. If no ℓ_{max} can be selected from $\mathcal{L}(\sigma_{max})$ that would decrease the maximum cost of sectors affected by $\mathcal{L}(\sigma_{max})$, sector σ_{max} is in the local minimum. In this case LRM moves to the sector with the next highest cost. Finally, after no further reduction in costs of the sectors is possible, the algorithm terminates, having found a locally optimal solution within the parameters of the search space. See the pseudocode in Figure 3.

We have developed GEOSECT-LOCAL, a tool implementing the LRM. As an input it takes a seed sectorization, scheduled traffic information, dominant flows and critical points, and a weather prediction. As an output, it produces a locally optimal sectorization with respect to the objective function. We use GEOSECT (presented in [12]) to extract dominant flows and critical points from the traffic schedule.

3.1 Local Adjustments

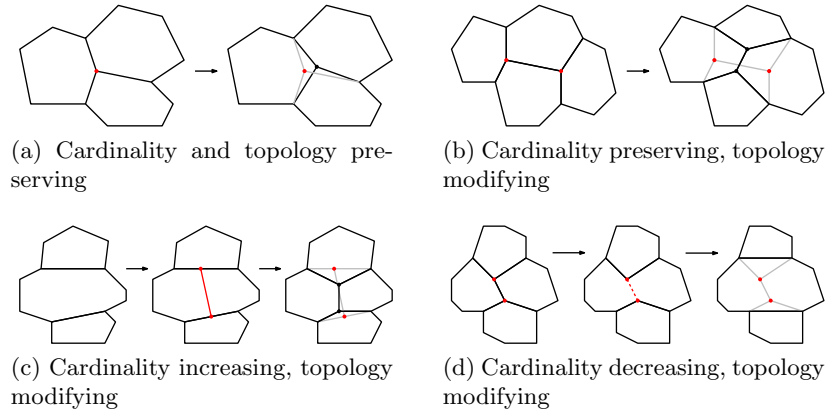


Fig. 4: Examples of local adjustments. (a) and (b) are cardinality preserving; (c) and (d) are not.

We distinguish between different types of local adjustments to an airspace partition. We say that a set of local adjustments is cardinality preserving if the number of sectors remains the same after the adjustments. Cardinality preserving adjustments can be of two subtypes: topology preserving (the graph defining the partition remains constant, while the positions of the vertices may be changed), and topology modifying (allowing structural changes to the graph of sector boundaries). Figure 4a is an example of a topology preserving adjustment, and Figure 4b is an example of a topology modifying adjustment. Local adjustments that are not cardinality preserving are necessarily topology modifying (since the number of faces of the planar graph changes, implying a change in topology); such adjustments are of two subtypes: merge (two or more sectors are merged into one) and split (one or more sectors are split into a greater number of sectors). A simple example of a split operation is the partitioning of a sector into two or into three smaller sectors; after the split, the newly added vertices (each of degree 3) are readjusted, as in Figure 4c. (Such partitions form the basis of GEOSECT’s top-down recursive partitioning algorithm.) A simple example of a merge operation is the deletion of a shared boundary uv in Figure 4d, between two adjacent sectors; after deletion, vertices u and v (now each of degree 2, since they lie along sector boundaries) are adjusted so that the corresponding sector boundaries are reoptimized.

We implement two types of local adjustments in GEOSECT-LOCAL: “vertex move” and “edge flip”. These are cardinality preserving local adjustments (refer to Figure 4). For “vertex move”, we identify all possible locations for relocating a vertex v using a grid centered around v . (Users can define the size and resolution of the search grid within the GUI.) The optimization evaluates the objective function at each candidate, and selects the candidate providing the lowest cost. For “edge flip” adjustment of edge e , we investigate edges

that are perpendicular to e and cross it in the middle. (Again, users can define the lengths and number of candidate edges.) While there are inefficiencies in this relatively brute-force approach to local optimization of vertex and edge position, using a search grid in this manner makes the testing of various objective functions straightforward and is consistent with the expectation that future airspace configurations will be based on an established grid of coordinates, the National Referencing System (NRS). We have tested some methods for reducing the number of discrete points searched for the optimal relocation, taking advantage of properties of the objective functions. However, these methods have had limited success. We have realized some computational efficiencies by evaluating components of the objective function only when necessary. Future work will address algorithm efficiency through selective search techniques, using bounding techniques to prune the search or using direct (non-grid) optimization techniques to optimize objective functions that are amenable to exact solutions.

3.2 Details of the Objective Function

As we have mentioned before, we describe a set of elemental parameters that measure the “quality” of a sectorization. For each of the parameters p we introduce one (or more) constraint c and associate a penalty function $\text{cost}_c(p)$. The penalty function $\text{cost}_c(p)$ gives a measure of how far the parameter p is outside of the permissible domain of values defined by c . For example, for parameter α (sector angle) one constraint can be to limit it from above by α_{max} to ensure that there will be no angles in sectors that are too large. Thus, $\text{cost}_c(\alpha)$ measures how far α is above α_{max} .

Figure 5 specifies all the constraints that are implemented in GEOSECT-LOCAL. The definitions of the most of the parameters presented in the table are intuitive, however some of them require a discussion.

Maximum and time-average aircraft count. Let $AC(\sigma, t)$ be the number of aircraft in sector σ at a time moment t . Then we can define the maximum aircraft count and the time-average aircraft count as

$$AC_{max}(\sigma) = \max_t(AC(\sigma, t)), \quad (2)$$

$$AC_{avg}(\sigma) = \frac{\int AC(\sigma, t) dt}{\int dt}, \quad (3)$$

Estimated delay. Let $K(\sigma)$ be the capacity of σ , *i.e.*, the number of aircrafts that air traffic controllers can safely operate in sector σ . The capacity of the sector can be estimated using one of the following methods. One simple measure of K commonly used is sector’s MAP value, estimated by $(5/3)$ times the average dwell time (in minutes). A more sophisticated estimate is that of [13], which defines the maximum allowed aircraft count to be

$$K = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad (4)$$

where $a = 6.8/V$, $b = a + 0.025 + 7/T$, $c = 0.7$, T is the average dwell time (in seconds), and V is the sector volume (in cubic nmi). Now we can define an estimated the delay as

$$\delta(\sigma) = \int (AC(\sigma, t) - K(\sigma)) dt. \quad (5)$$

Sector convexity. We use a simple measure of non-convexity $cx(\sigma)$: the ratio of the area of sector σ to the area of its convex hull, *i.e.*,

$$cx(\sigma) = \frac{\text{area}(\sigma)}{\text{area}(\text{ConvexHull}(\sigma))}. \quad (6)$$

If this ratio is 1, the sector is convex. Smaller values of the ratio correspond to a greater degree of non-convexity.

#	parameter	parameter description	constraint	default threshold	parameter limit
1	AC_{avg}	time-average airplane count	$AC_{avg} \leq T_1$	T_1	$L_1 = \infty$
2	AC_{max}	max airplane count	$AC_{max} \leq T_2$	T_2	$L_2 = \infty$
3	δ	estimated delay	$\delta \leq T_3$	$T_3 = 0$	$L_3 = \infty$
4	N_L	throughput	$N_L \geq T_4$	$T_4 = 2$	$L_4 = 0$
5	T_{dwell}	dwel time	$T_{dwell} \geq T_5$	$T_5 = 300 \text{ sec}$	$L_5 = 0 \text{ sec}$
6	β	DF-boundary crossing angle	$\beta \leq \beta_{max}$	$T_6 = \beta_{max} = 45^\circ$	$L_6 = 90^\circ$
7	d_{DF}	DF-boundary distance	$d_{DF} \geq T_7$	$T_7 = 0.3'' \text{ (long/lat)}$	$L_7 = 0''$
8	d_{CP}	CP-boundary distance	$d_{CP} \geq T_8$	$T_8 = 0.4'' \text{ (long/lat)}$	$L_8 = 0''$
9	α	sector angle	$\alpha \geq \alpha_{min}$	$T_9 = \alpha_{min} = 80^\circ$	$L_9 = 0^\circ$
10	α	sector angle	$\alpha \leq \alpha_{max}$	$T_{10} = \alpha_{max} = 180^\circ$	$L_{10} = 360^\circ$
11	cx	sector convexity	$cx \geq cx_{min}$	$T_{11} = cx_{min} = 90\%$	$L_{11} = 0\%$
12	$ e $	edge length	$ e \geq T_{12}$	$T_{12} = 0.4'' \text{ (long/lat)}$	$L_{12} = 0''$

Fig. 5: Description of constraints, corresponding parameters used in LRM, and default settings for GEOSCT-LOCAL.

Throughput. We define a throughput value N_L in a sector σ along a dominant flows df to be the number of air traffic lanes admissible by σ along the flow df . The sectors with the throughput 2 or more provide additional alternate lanes that can be used for rerouting the traffic if needed. We compute N_L by using the max-flow/min-cut analysis (refer to [14–16]).

Note that some of the objectives are defined either with respect to the set of all tracks (trajectories in the input data) or with respect to a set of *dominant flows*, or route structures, given as input. The dominant flows represent the primary flows, or route structures, of traffic across the airspace of interest. Dominant flows are dynamic, and need to be updated over time with changing conditions in traffic, especially in cases of weather-impacted airspace. For related work on dominant flow extraction, see [12] and [5].

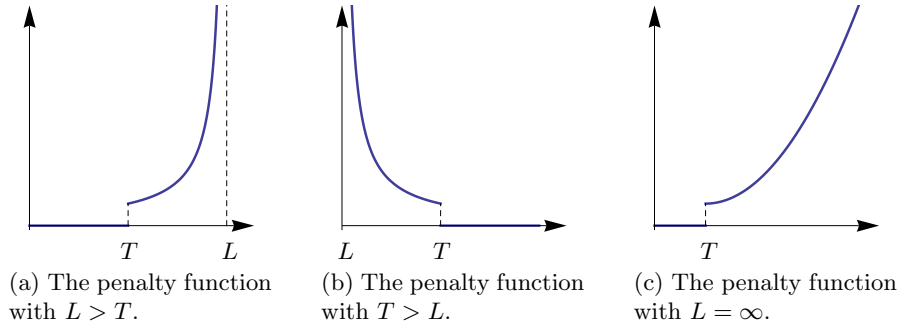


Fig. 6: The penalty function.

The formulas of penalty functions $\text{cost}_c(\sigma)$ depend on two constants T_c and L_c , where T_c is a user-defined threshold for parameter p , and L_c is a physical limit for p (refer to Figure 5). For example, the upper limit of a sector angle is 360° (sector angles cannot physically be greater than 360°), the lower limit of an edge length is 0 (edges cannot have negative length), the upper limit for the average aircraft number in a sector is ∞ (technically, there is no limit on the number of airplanes in a sector). For the constraints on the parameters with bounded limit L_c (such as constraints on sector angles, distances bounded from below, etc.) the penalty function is the following (refer to Figures 6a and 6b):

$$\text{cost}_c(p) = \begin{cases} \frac{T_c - L_c}{p - L_c}, & \text{if } p \text{ is between } T_c \text{ and } L_c, \\ 0, & \text{otherwise.} \end{cases}$$

For the constraints on parameters with $L_c = \infty$ (see Figure 6c):

$$\text{cost}_c(p) = \begin{cases} (p - T_c)^2 + 1, & \text{if } p \geq T_c, \\ 0, & \text{otherwise.} \end{cases}$$

The choice of the objective function used in evaluating the quality of each of the candidate local adjustments is critical to the performance of the local optimization. One approach to multi-criteria optimization is to combine all the criteria into a single objective function, $\text{cost}(\sigma) = \sum (w_c \text{cost}_c(\sigma))$, according to user-specified weights w_c . We chose this approach in GEOSECT-LOCAL as the most natural one.

Another approach would be to treat the individual components as a vector, (c_1, c_2, \dots, c_k) , and optimize sets of components while constraining other components, *e.g.*, to obtain Pareto-optimal solutions.

4 Experimental Results

We admit that in the worst case LRM can produce a sectorization that is arbitrarily bad in comparison with the optimal sectorization. For example, Figure 7 shows two toy situations when LRM can get “stuck” in a local minimum of the objective function. Left example shows a sectorization in a local minimum for a case when the weight of the penalty function of the convexity constraint is high enough in order not to allow appearance of non-convex sectors. Right example shows a case of a non-convex region that does not allow a single local adjustment to be made, and thus, is stuck in a local minimum.

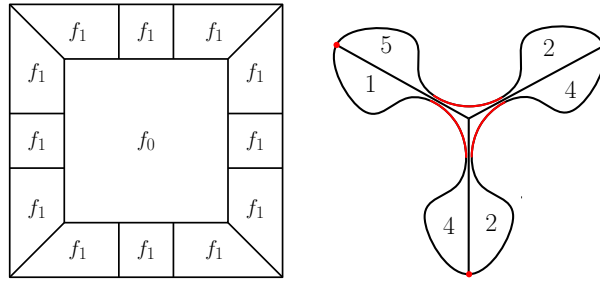


Fig. 7: Bad examples when GEOSECT-LOCAL can get stuck. Left: if $f_0 < f_1$ are current sectors workloads, GEOSECT-LOCAL cannot make a local adjustment while keeping sectors convex. Right: Let numbers in sectors denote the workload in the corresponding “corner”. Optimum solution would be to connect the center with the red areas, but impossible to achieve by singular local moves.

Despite the obvious possibility of an arbitrarily bad solution, in practice GEOSECT-LOCAL has proven to produce competitive sectorizations. In comparison with the sectorizations currently used by NAS GEOSECT-LOCAL sectorizations improve the workload balance among sectors and reduce sector delays, in some cases by 50% and more.

4.1 Synthetic Experiments

We have run a number of synthetic experiments to analyze the advantages and the drawbacks of the LRM. The test suit consisted of a number of randomly generated experiment settings consisting of a set of airports uniformly distributed in a $12'' \times 12''$ square. Each airport had a corresponding normally distributed random weight which represented the size of the airport and affected the amount of traffic departing and landing at the airport. For each experiment setting we generated several random weather scenarios and sets of trajectories. The region that we chose to sectorize was a square of size $9'' \times 9''$ in the center of the $12'' \times 12''$ square. This way there was traffic passing through the region as well as fully contained inside it. Figure 8a shows one of the experiment settings with a weather scenario and a traffic generated for this scenario. Figures 8b and 8c show the region of interest with dominant flows extracted with the use of GEOSECT, and a sample seed sectorization. Finally, Figure 8d shows the output GEOSECT-LOCAL sectorization.

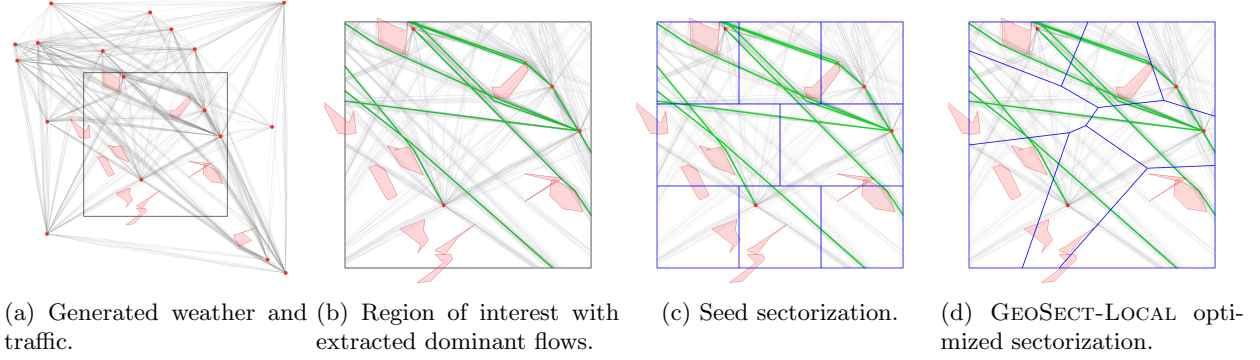


Fig. 8: Example of a synthetic experiment setting.

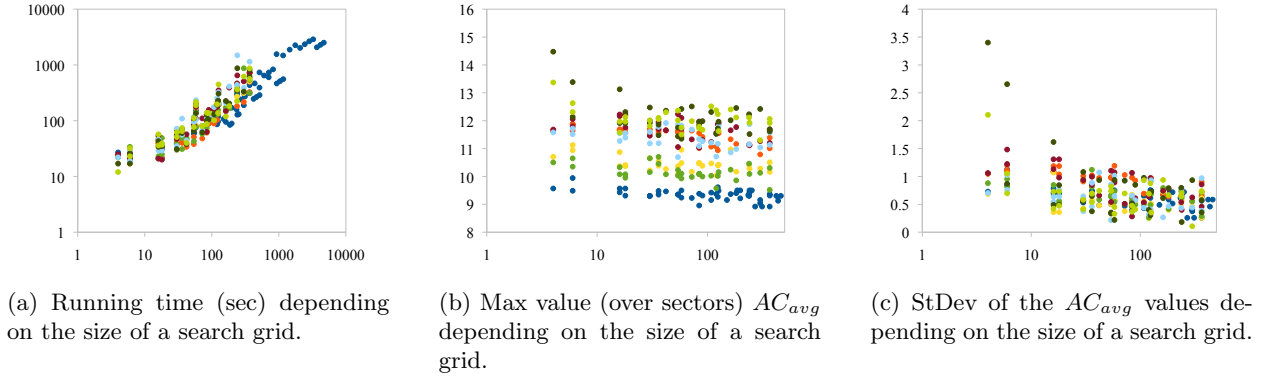


Fig. 9: Dependency of the running time and resulting AC_{avg} on the grid size.

The first round of experiments tested the dependency of the running time of the LRM and the variation of AC_{avg} on the size of a search grid. The charts in Figure 9 display the results for 8 different experiment settings. We can see from these charts that the running time of the LRM nearly linearly depends on the size of the search grid, but there is not much of a gain in balancing the AC_{avg} for grids with more than 30 points. Based on this we chose the search grid to have $0.4''$ radius and $0.15''$ grid step (in total 36 points) for the following experiments.

	Worst AC_{avg}		Min convexity	
	AVG	StDev	AVG	StDev
No weather	8.1	0.11	0.99	0.03
Weather 1	9.55	0.11	0.99	0.04
Weather 2	9.06	0.07	1	0
Weather 3	8.74	0.06	0.99	0.03

Fig. 10: Dependency between the workload and the convexity.

We have run a number of tests to determine the correlation between the workload parameters and the geometric parameters. Figure 10 shows a summary of an experiment where the AC_{avg} was minimized, using various weight settings, under the convexity constraint. We observe that with no other constraints than the convexity constraint GEOSect-LOCAL produces sectorizations that are very close to optimal for any weight values.

We have also run similar experiments by selecting three other constraints to serve as a counterpart to the AC_{avg} parameter: constraint on the maximum intersection angle of dominant flows and sector boundaries,

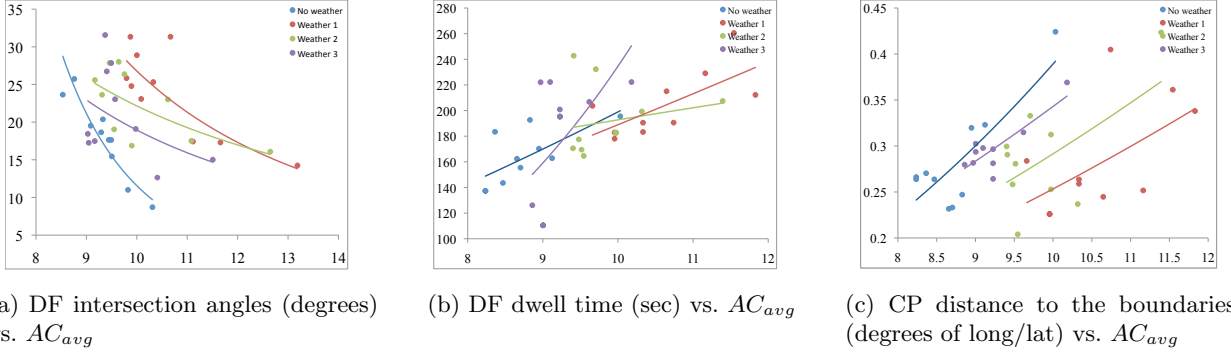


Fig. 11: Dependency between the workload and geometric parameters of sectorizations.

constraint on the minimum dwell time, and constraint on the minimum distance between critical points and the boundaries. Figure 11 shows the dependencies between the AC_{avg} and these geometric parameters, that could be predicted: the smaller the average AC_{avg} the higher intersection angles of the flows with the boundaries and the smaller minimum dwell time and the distance from the critical points to the boundaries.

4.2 Historical Data Experiments

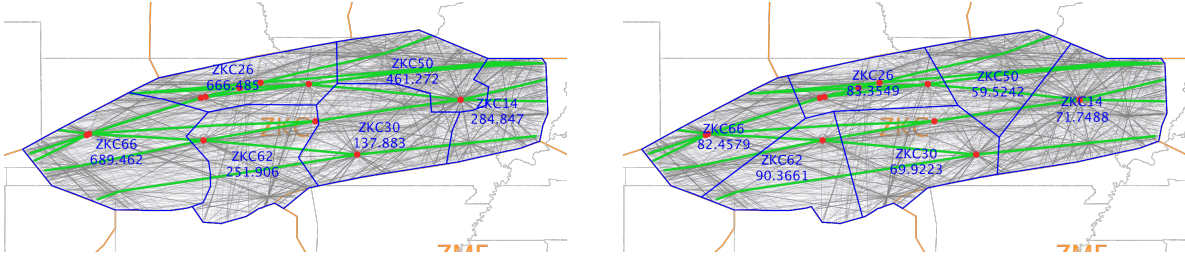


Fig. 12: Baseline sectorization before and after rebalancing. Numbers in sectors show the total sector cost.

	Parameter Value		Cost					
	Worst		Avg		Max		StDev	
	before	after	before	after	before	after	before	after
min sec \angle	38°	59°	4.3	2.9	13.6	5.8	3.6	1.6
max sec \angle	278°	191°	4.2	1.6	11.7	1.6	3.2	0
convexity cx	0.85	0.95	1.5	0	1.5	0	0.03	0
dist to CPs	0.08"	0.4"	35.5	7.3	138.7	9.6	51.0	1.7
dist to DFs	0.08"	0.25"	49.9	12.5	103.1	20.0	30.0	5.2
max \angle w/ DFs	76°	65°	23.1	9.8	64.2	23.8	24.1	8.2
min dwell	42 sec	125.5 sec	51.9	40.5	167.1	40.5	77.2	0
est. delay	20.4 min	15.3 min	105.4	35.4	278.9	62.7	104.5	23.3
total cost:			415.3	76.2	689.5	90.4	228.5	11.2

Fig. 13: Comparisons of the parameters and penalties of the baseline sectorization before and after rebalancing.

In the experiment presented in this section we concentrate on minimizing the average delay in the Kansas City (ZKC) center for 36 hours of historical traffic data. We set the priority of the optimization to be the estimated delay by putting higher weight on the corresponding penalty function and lower weights on other constraints. As an input sectorization to GEOSECT-LOCAL we selected the current NAS sectors for the ZKC center. Fig. 12 shows the baseline sectorization before and after rebalancing.

Figure 13 compares parameters’ penalties for the baseline sectorization (before and after rebalancing). There are improvements in average and maximum values as well as in standard deviation of penalties for all the parameters. GEOSECT-LOCAL improved the average delay over all sectors and reduced the maximum delay by 25%.

4.3 Simulation Experiment

In collaboration with NASA Ames and Metron Aviation, we conducted experiments utilizing NASA’s ACES Flight simulator. The flight simulator takes scheduled flights and a sectorization as an input and generates flight trajectories.

The goal of the experiment was to evaluate the sectorizations produced by LRM in “real-life” settings of ZKC center. The experiment had 3 stages corresponding to 3 time periods in a day: early morning (light traffic), early afternoon (the heaviest traffic), and evening (normal traffic). There are 3 current baseline sectorizations corresponding to these time intervals consisting of 6, 24 and 19 sectors respectively. A set of sample trajectories (for flights schedule with twice the usual demand) was generated by the simulator for the first stage with the option no input sectorization. Based on this projected flight trajectories for the first time interval we were to produce a sectorization and feed it in the flight simulator for the generation of the flight trajectories for the second stage of the experiment. The output sectorization of the second stage was used in the same way to generate the trajectories for the third stage.

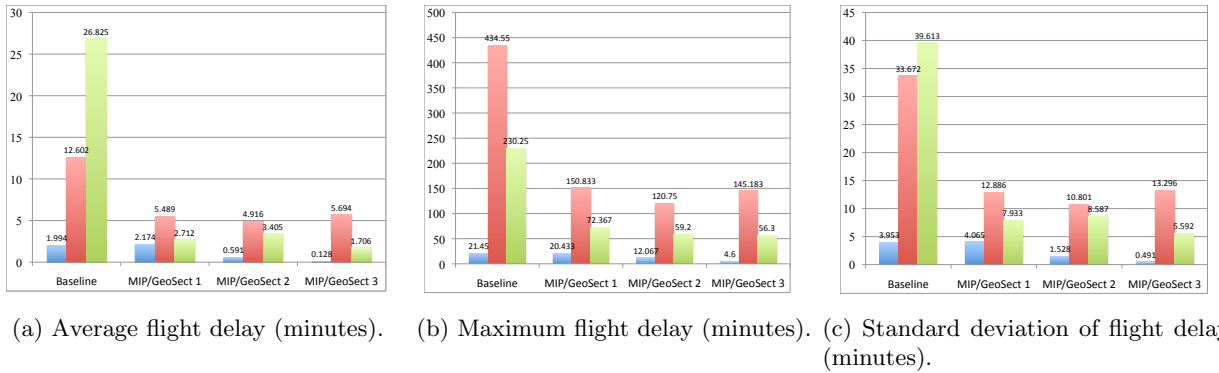


Fig. 14: Comparison of flight delays induced by baseline sectorizations and three sets of MIP/GeoSect sectorizations.

Our colleagues at Metron Aviation used the Mixed Integer Programming (MIP) method [2] to generate the seed sectorizations that were processed with GEOSECT-LOCAL. Our objective was to minimize the estimated delay and to compare how well it corresponds to the delay computed by the NASA’s ACES Flight simulator. We have generated three different sets of sectorizations with the same number of sectors or less than in the baseline sectorizations. The comparison of the delays in the resulting MIP/GEOSECT-LOCAL sectorizations is presented in Figure 14. MIP/GEOSECT-LOCAL approach was able to almost eliminate the average delay in the first stage, and reduce it in 2.5-7.8 times for the second and third stages.

5 Conclusion

We have studied the ASP problem, reviewed its complexity and presented a heuristic LRM that uses a multi-criteria optimization approach to improve an input sectorization.

We have implemented LRM algorithms in GEOSECT-LOCAL, a highly configurable tool that produces high quality sectorizations. The result depends on the choice of initial sectorizations (more specifically on their topology, and to a less degree on their exact geometry). In all the experiments GEOSECT-LOCAL significantly reduced the average aircraft count and delay, and the sectors had “nice” geometric and flow conforming

properties. One of the important advantages of the LRM is that it can optimize sectorizations with respect to any constraint that can be described with a simple parameter that can be evaluated numerically. Thus, LRM is not limited to the set of constraints described in this paper.

Current and future work include extending LRM to produce dynamic sectorizations that continuously adapt to the changing traffic, and to produce robust sectorizations under uncertain traffic patterns.

Acknowledgment

This work is supported by NASA Ames research Center under contracts number NNA07BB33C and number NNA11AB99C.

We thank our colleagues Girishkumar Sabhnani and Arash Yousefi from Metron Aviation for useful discussions and domain expertise on air traffic management. We also thank Metron Aviation for providing us with the generated sectorizations, and historical and generated track data.

References

1. P. Lee, J. Mercer, B. Gore, N. Smith, K. Lee, and R. Hoffman, "Examining airspace structural components and configuration practices for dynamic airspace configuration," in *AIAA Guidance, Navigation, and Control Conference*, Honolulu, HI, 2008.
2. A. Yousefi and G. Donohue, "Optimum airspace sectorization with air traffic controller workload constraints," in *1st International Conference for Research in Air Transportation*, Zilina, Slovakia, November 2004.
3. A. Yousefi, "Optimum airspace design with air traffic controller workload-based partitioning," Ph.D. dissertation, George Mason University, 2005.
4. A. Basu, J. S. B. Mitchell, and G. Sabhnani, "Geometric algorithms for optimal airspace design and air traffic controller workload balancing," *ACM Journal of Experimental Algorithmics*, vol. 14, pp. 2.3–2.28, 2009.
5. G. Sabhnani, A. Yousefi, and J. Mitchell, "Flow conforming operational airspace sector design," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Fort Worth, TX, September 2010.
6. M. Xue, "Airspace sector redesign based on voronoi diagrams," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 12, pp. 624–634, December 2009.
7. S. Martinez, G. Chatterji, D. Sun, and A. Bayen, "A weighted-graph approach for airspace dynamic configuration," in *AIAA Conference on Guidance, Navigation and Control*, August 2007.
8. A. Klein, M. Rodgers, and H. Kaing, "Dynamic fpas: A new method for dynamic airspace configuration," in *Integrated Communications, Navigation and Surveillance Conference, 2008. ICNS 2008*, may 2008, pp. 1–11.
9. M. Drew, "Dynamically evolving sectors for convective weather impact," in *AIAA 2010-9161, 10th AIAA Aviation*, 2010.
10. M. Bloem and P. Kopardekar, "Combining airspace sectors for the efficient use of air traffic control resources," in *AIAA Guidance, Navigation, and Control Conference*, August 2008.
11. A. H. Farrahi and Z. Wood, "Computational complexity of the airspace sectorization problem."
12. G. Sabhnani, A. Yousefi, D. P. Kierstead, V. Polishchuk, J. S. B. Mitchell, and I. Kostitsyna, "Algorithmic traffic abstraction and its application to nextgen generic airspace," in *10th AIAA Aviation Technology Integration and Operations ATIO Conference*, September 2010, pp. 2093–2102.
13. J. D. Welch, J. W. Andrews, B. D. Martin, B. Sridhar, and M. Field, "Macroscopic workload model for estimating en route sector capacity," *7th USA/Europe Air Traffic Management Research and Development Seminar ATM2007*, pp. 1–10, 2007.
14. J. S. B. Mitchell, "On maximum flows in polyhedral domains," *Journal of Computer and System Sciences*, vol. 40, pp. 88–123, 1990.
15. J. S. B. Mitchell, V. Polishchuk, and J. Krozel, "Airspace throughput analysis considering stochastic weather," in *AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.
16. J. S. B. Mitchell and V. Polishchuk, "Thick non-crossing paths and minimum-cost flows in polygonal domains," in *Proc. 23rd Annual Symposium on Computational Geometry*, 2007, pp. 56–65.